S18	319	"scalable vector graphics" "svg" Adj statement$	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 14:39
S19	319	"scalable vector graphics" "svg" Adj statement$5 & document$	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 18:29
S20	260	S19 & database	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 14:39
S21	227	S20 & (link hyperlink$)	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 14:48
S22	1	"svg statement$"	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 14:48
S23	0	"scalable vector graphics statement$"	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 14:48
S24	8	svg same statement$	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 14:49
S25	1	"svg statement$"	US-PGPUB; USPAT	OR	OFF	2004/12/07 17:01
S26	5	"scalable vector graphics" same statement$	US-PGPUB; USPAT	OR	OFF	2004/12/07 17:01
S27	76	sall.in.	US-PGPUB; USPAT	OR	OFF	2004/12/07 17:25
S28	319	"scalable vector graphics" "svg" Adj statement$5 & document$ & dtd	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/07 18:58
S29	27	("scalable vector graphics statement$" SVG) & document & (link hyperlink) & GIS	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/07 18:59
S30	212	("scalable vector graphics statement$" SVG) & document & (link hyperlink) & map	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/08 09:21
S31	248	svg & query	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/08 09:25

S32	0	"svg statement" & query	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/12/08 09:27
S35	1682	"geographical information system" GIS	US-PGPUB; USPAT	OR	OFF	2004/12/08 09:43
S36	31	S35 & svg	US-PGPUB; USPAT	OR	OFF	2004/12/08 09:47
S37	805	database same "network devices"	US-PGPUB; USPAT	OR	OFF	2004/12/08 09:48
S38	2	S37 & svg	US-PGPUB; USPAT	OR	OFF	2004/12/08 09:48
S39	9	dtd & svg & "style sheet$"	US-PGPUB; USPAT	OR	OFF	2004/12/08 10:42
S40	47	dtd & svg	US-PGPUB; USPAT	OR	OFF	2004/12/08 11:08
S41	6877	"network topology"	US-PGPUB; USPAT	OR	OFF	2004/12/08 11:09
S42	19	S41 & svg	US-PGPUB; USPAT	OR	OFF	2004/12/08 11:09
S43	17	("4977520" | "5051736" | "5220657" | "5477012" | "5486686" | "5652412" | "5661506" | "5692073" | "5852434" | "5862321" | "5869819" | "6076734" | "6081261" | "6460766" | "6466955" | "6628847" | "6666376").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2004/12/09 20:04
S44	1235	"scalable vector graphics" SVG	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S45	6	("scalable vector graphics" SVG) & "graphical interface"	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S46	577	("scalable vector graphics" SVG) & document	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S47	199	(("scalable vector graphics statement$" SVG) & document) & graphical & object	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S48	130	((("scalable vector graphics statement$" SVG) & document) & graphical & object) & query	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S49	571	("scalable vector graphics statement$" SVG) & document	US-PGPUB; USPAT	OR	OFF	2005/08/02 15:49
S50	456	("scalable vector graphics statement$" SVG) & document & (link hyperlink)	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19

S51	456	("scalable vector graphics statement$" SVG) & document & (link hyperlink)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/02 14:19
S52	25	udelle.in.	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S53	0	thomas-kas.in.	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S54	1	("6549935").URPN.	USPAT	OR	OFF	2005/08/02 14:19
S55	8	("5006991" | "5615321" | "5619630" | "5815154" | "5821941" | "5963212" | "5999186" | "6219055").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S56	2	svg.ti.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S57	8	aptus.in.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S58	1	"scalable vector graphics statement$" "svg statement$" same document$	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S59	2	"scalable vector graphics statement$" "svg statement$"	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S60	478	"scalable vector graphics" "svg" Adj statement$	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S61	478	"scalable vector graphics" "svg" Adj statement$5 & document$	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S62	382	S61 & database	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S63	339	S62 & (link hyperlink$)	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S64	2	"svg statement$"	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S65	0	"scalable vector graphics statement$"	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S66	12	svg same statement$	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19

S67	2	"svg statement$"	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S68	7	"scalable vector graphics" same statement$	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S69	91	sall.in.	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S70	478	"scalable vector graphics" "svg" Adj statement$5 & document$ & dtd	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S71	33	("scalable vector graphics statement$" SVG) & document & (link hyperlink) & GIS	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/02 14:19
S72	320	("scalable vector graphics statement$" SVG) & document & (link hyperlink) & map	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/02 14:19
S73	363	svg & query	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/02 14:19
S74	0	"svg statement" & query	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/02 14:19
S75	1990	"geographical information system" GIS	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S76	39	S75 & svg	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S77	971	database same "network devices"	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S78	2	S77 & svg	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S79	12	dtd & svg & "style sheet$"	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S80	59	dtd & svg	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S81	8117	"network topology"	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19
S82	19	S81 & svg	US-PGPUB; USPAT	OR	OFF	2005/08/02 14:19

S83	17	("4977520" | "5051736" | "5220657" | "5477012" | "5486686" | "5652412" | "5661506" | "5692073" | "5852434" | "5862321" | "5869819" | "6076734" | "6081261" | "6460766" | "6466955" | "6628847" | "6666376").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 14:19
S84	1005	xml same graphics	US-PGPUB; USPAT	OR	OFF	2005/08/02 15:50
S85	208	S84 & "vector graphics"	US-PGPUB; USPAT	OR	OFF	2005/08/02 15:50
S86	17	("4710763" | "5247651" | "5610653" | "5619632" | "5701137" | "5748186" | "5805171" | "5819243" | "5893109" | "5956736" | "6055544" | "6144962" | "6189019" | "6219053" | "6337702" | "6340977" | "6356902").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/02 15:58
S87	71	(svg "vector graphic$") & (dtd "document type definition")	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:19
S88	141	donoho.in.	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:20
S89	0	donoho-andrew.in.	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:19
S90	0	S88 & svg	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:19
S91	1180	cohn.in.	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:20
S92	0	S91 & svg	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:20
S93	1380	bowler.in. svg	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:20
S94	0	bowler.in. & svg	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:21
S95	0	bowler.in. & "vector graphics"	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:26
S96	35	northway.in.	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:27
S97	2	"svg statement$"	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:28
S98	80	(svg "vector graphics") & dtd	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:29
S99	59	(svg) & dtd	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:35

S100	0	"svg statement$" & dtd	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:35
S101	2	"svg statement"	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:39
S102	317	svg & hyperlink$	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:40
S103	293	S102 & database	US-PGPUB; USPAT	OR	OFF	2005/08/03 09:40
S104	293	S102 & database	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:40
S105	502	svg & database	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:40
S106	293	S105 & hyperlink$	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:43
S107	31	"svg document"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:46
S108	1510	("scalable vector graphics" svg)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:46
S109	199	S108 same document$	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:47
S110	32	S109 & dtd	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:51

S111	45449	"svg" editor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 09:51
S112	18	"svg" same editor	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 10:41
S113	277	"svg" & hyperlink & database	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 10:44
S114	240	"svg" same hyperlink & database	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 11:52
S115	256	sall.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 11:52
S116	0	sall.in. & svg	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/08/03 11:52
S117	3	("6611725").URPN.	USPAT	OR	OFF	2005/08/03 11:59
S118	14123	CAD	USPAT	OR	OFF	2005/08/03 11:59
S119	2937	S118 & database	USPAT	OR	OFF	2005/08/03 11:59
S120	10	S119 & svg	USPAT	OR	OFF	2005/08/03 11:59

previous next contents properties index

19 Embedding Foreign Object Types

(This section is still under construction.)

One goal for SVG is to provide a mechanism by which other XML language processors can render into an area within an SVG drawing, with those renderings subject to the various transformations and compositing parameters that are currently active within the SVG document. One particular example of this is to provide a frame for the HTML/CSS processor so that dynamically reflowing text (subject to SVG transformations and compositing) could be inserted into the middle of an SVG document. Another example is inserting a MathML expression into an SVG drawing.

This facility would include a capability for some sort for an alternative representation of the embedded foreign object so that something meaningful might appear in SVG viewing environments which do not have the ability to process the embedded foreign object. One way this might work would be via a **<switch>** element and **system-required** attribute similar to the corresponding facilities within the SMIL 1.0 Recommendation.

To illustrate the desired capabilities with an example, we'll assume that:

- SVG has an element named **<foreignobject>** which permits non-SVG elements within itself
- SVG has a **<switch>** element that works like the same-named element in SMIL 1.0 (i.e., the first child element whose testing attributes evaluate to true is processed and all other child elements are skipped)
- SVG's **<foreignobject>** element has a **system-required** attribute that works like the same-named attribute in SMIL 1.0 (i.e., evaluates to true if the user agent supports the given named extension or capability)

```
<?xml version="1.0" standalone="yes"?>
<svg width="4in" height="3in"
 xmlns = 'http://www.w3.org/Graphics/SVG/svg-19990412.dtd'>
  <desc>This example uses the switch element to provide a
  fallback graphical representation of an equation, if
  MathML is not supported.
  </desc>
  <!-- The <switch> element will process the first child element
       whose testing attributes evaluate to true.-->
  <switch>
```

```
    <!-- Process the MathML if the system-required attribute
         evaluates to true (i.e., the user agent supports MathML
         embedded within SVG). -->
    <foreignobject
       system-required="http://www.w3.org/TR/REC-MathML-19980407"
       width="100" height="50">
      <!-- MathML content goes here -->
    </foreignobject>

    <!-- Else, process the following alternate SVG.
         Note that there are no testing attributes on the <g> element.
         If no testing attributes are provided, it is as if there
         were testing attributes and they evaluated to true.-->
    <g>
      <!-- Draw a red rectangle with a text string on top. -->
      <rect style="fill: red"/>
      <text>Formula goes here</text>
    </g>

  </switch>
</svg>
```

Download this example

It is not required that SVG processors support the ability to invoke other arbitrary processors to handle embedded foreign object types; however, all conforming SVG processors would need to support the **<switch>** element and should be able to render valid SVG elements when they appear as one of the alternatives within a **<switch>** element.

It is expected that commercial Web browsers at a minimum will support the ability for SVG to embed content from other XML grammars which use CSS layout or XSL to format their content, with the resulting CSS- or XSL-formatted content subject to SVG transformations and compositing.

(The exact mechanism for providing these capabilities hasn't been decided yet. Many details need to be worked out.)

next contents properties index

WD-SVG-19990412

Scalable Vector Graphics (SVG) Specification

W3C Working Draft *12 April 1999*

This version: http://www.w3.org/TR/1999/WD-SVG-19990412/
Latest version: http://www.w3.org/TR/WD-SVG/
Previous version: http://www.w3.org/TR/1999/WD-SVG-19990326/

Editor: Jon Ferraiolo <jferraio@adobe.com>

Authors: John Bowler, Microsoft Corporation <johnbo@microsoft.com>
Richard Cohn, Adobe Systems Incorporated <cohn@adobe.com>
Milt Capsimalis, Autodesk Inc. <milt@autodesk.com>
Andrew Donoho, IBM <awd@us.ibm.com>
David Duce, RAL (CCLRC) <dad@inf.rl.ac.uk>
Jerry Evans, Sun Microsystems <jerry.evans@Eng.sun.com>
Jon Ferraiolo, Adobe Systems Incorporated <jferraio@adobe.com>
Scott Furman, Netscape Communications Corporation <fur@netscape.com>
Peter Graffagnino, Apple <pgraff@apple.com>
Lofton Henderson, Inso Corporation <lofton@cgm.com>
Alan Hester, Xerox Corporation <Alan.Hester@usa.xerox.com>
Bob Hopgood, RAL (CCLRC) <frah@inf.rl.ac.uk>
Kelvin Lawrence, IBM <klawrenc@us.ibm.com>
Chris Lilley, W3C <chris@w3.org>
Philip Mansfield, Inso Corporation <philipm@paradigmdev.com>
Kevin McCluskey, Netscape Communications Corporation <kmcclusk@netscape.com>
Tuan Nguyen, Microsoft Corporation <tuann@microsoft.com>
Troy Sandal, Visio Corporation <TroyS@visio.com>
Peter Santangeli, Macromedia <psantangeli@macromedia.com>
Haroon Sheikh, Corel Corporation <haroons@corel.ca>
Gavriel State, Corel Corporation <gavriels@COREL.CA>
Robert Stevahn, Hewlett-Packard Company <rstevahn@boi.hp.com>

Shenxue Zhou, Quark <szhou@quark.com>

Status of this document

This document is an intermediate public review draft version of the SVG specification.

The SVG working group has been using a staged approach. Initially, the working group developed an detailed set of SVG Requirements, which are listed in Appendix A:. These requirements were posted for public review initially in November 1998. For the most part, the specification has been developed to provide the feature set listed in the requirements document. Appendix A contains detailed editorial comments about which requirements have been addressed in this draft (along with hyperlinks to the relevant sections of the specification) and notes about which requirements have not been addressed yet and why.

The SVG working group has achieved significant progress toward translating the SVG requirements into an SVG specification. Major sections of the SVG language have either a proposed initial syntax and/or an in-depth description of the direction the working group is considering. However, there is still much work to be done. The working group still needs to perform in-depth review of many sections of this document. Other sections have yet to be written up at all, even in preliminary form. There is a need for considerable coordination work with other W3C working groups. Overall, it is likely that significant changes to the SVG specification will occur before a Proposed Recommendation is delivered by the working group.

Despite the preliminary nature of this draft specification, tools vendors and Web content creators are encouraged to experiment and develop preliminary versions of tools and Web sites according this draft specification, with the understanding that these tools and Web sites are experiemental/developmental in nature only and will need to be adapted to the final SVG Recommendation.

The main goal with this draft specification is to solicit public review and feedback. Public discussion of SVG features takes place on www-svg@w3.org, which is an automatically archived email list. Information on how to subscribe to public W3C email lists can be found at http://www.w3.org/Mail/Request. Review comments should be sent to www-svg@w3.org,

The home page for the W3C graphics activity is http://www.w3.org/Graphics/Activity.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Abstract

This specification defines the features and syntax for Scalable Vector Graphics (SVG).

SVG is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility.

SVG drawings can be dynamic and interactive. The Document Object Model (DOM) for SVG allows for straightforward and efficient vector graphics animation via scripting. A rich set of event handlers such as onmouseover and onclick can be assigned to any SVG graphical object. Because of its compatibility and leveraging of other Web standards, features like scripting can be done on HTML and SVG elements simultaneously within the same Web page.

Available formats

The SVG specification is available in the following formats. (In future versions, the specification's vector drawings will be available in both SVG and raster image formats. For now, only raster image formats are available.)

HTML4:
http://www.w3.org/TR/1999/WD-SVG/index.html

Available languages

The English version of this specification is the only normative version. However, for translations in other languages see http://www.w3.org/Graphics/SVG/svg-updates/translations.html.

Quick Table of Contents

The following sections have not been written yet, but are expected to be be present in later versions of this specification:

- **Appendix D. Sample SVG files**
- **Appendix E. Implementation and performance notes for fonts**
- **Appendix F. SVG's Document Object Model (DOM)**
- **Appendix G. SVG Support for XML Fragments**
- **Appendix H. Minimizing SVG File Sizes**
- **Appendix I. Internationalization Support**
- **Appendix J. Accessibility Support**
- **Appendix K. References**
- **Appendix L. Property and attribute index**
- **Appendix M. Index**

Full Table of Contents

The following sections have not been written yet, but are expected to be be present in later versions of this specification:

- **Appendix D. Sample SVG files**
- **Appendix E. Implementation and performance notes for fonts**
- **Appendix F. SVG's Document Object Model (DOM)**
- **Appendix G. SVG Support for XML Fragments**
- **Appendix H. Minimizing SVG File Sizes**
- **Appendix I. Internationalization Support**
- **Appendix J. Accessibility Support**
- **Appendix K. References**
- **Appendix L. Property and attribute index**
- **Appendix M. Index**

Copyright © 1999 W3C (MIT, INRIA, Keio), All Rights Reserved.

next contents properties index

NEWS.COM http://www.news.com/

W3C advances shrink-to-fit graphics technology

By Paul Festa
http://news.com.com/W3C+advances+shrink-to-fit+graphics+technology/2100-1023_3-243988.html

Story last modified Wed Aug 02 13:50:00 PDT 2000

More than a year behind schedule, a Web standards body today advanced a graphics technology aimed at making computer images fit into any screen--from cell phone displays to large monitors.

The World Wide Web Consortium (W3C) is inviting comment on Scalable Vector Graphics (SVG), advancing the technology to the standards body's penultimate "candidate recommendation" status.

The specification, first proposed in January last year and originally slated for a summer 1999 proposed recommendation, promises to make Web graphics more flexible and lightweight, as well as more easily integrated with Web documents.

Vector graphics are images that computers can render from a set of geometric descriptions instead of pixel-by-pixel bitmap copies such as the common JPEG or GIF formats. Because vector graphics are mere abstract descriptions, they can fly through tight bandwidth connections that typically choke on bulky image files.

Vector graphics also have the advantage of being easily resized to suit their destinations. On that score, SVG comes at an opportune time for companies tailoring Web pages to fit a variety of different Web-surfing devices, including small appliances such as telephones.

Because photographs still require bitmap formats, vector graphics don't spell the end of bitmap images on the Web. To that end, SVG is designed to assimilate and more efficiently resize photographic images, according to the W3C.

SVG is written in Extensible Markup Language (XML), a W3C recommendation for creating specialized markup languages for the Web. Capitalizing on XML's capabilities, the W3C has made SVG's textual content, such as logos and labels, searchable and translatable, among other things.

"Businesses have a lot of data in XML," said Chris Lilley, W3C graphics activity lead. "This is being used for data, for text. Putting the graphic in XML means it can be manipulated using the same tools and as a unit with the text. It's a much higher level of integration."

SVG emerged as a Tower of Babel threatened to rise above vector graphics technologists. Work on the specification began as an attempt to synthesize the development of numerous competing vector graphics submissions that had made their way to the W3C by early last year.

These included Web Schematics, designed for making flow charts and other diagrams; Adobe's PostScript-based Precision Graphics Markup Language (PGML), best suited to graphics such as bar charts, logos and screen graphics like push buttons; the Microsoft-backed Vector Markup Language (VML), a text format for vector graphics; and DrawML, which resembles Web Schematics except that it relies heavily on Java, Sun Microsystems' cross-platform programming language, to lay out diagrams.

But the SVG soon took a course away from the existing proposals, according to the W3C.

"We looked at those things and PostScript and even VRML," Lilley said. "But you shouldn't think this has been a pulling together of existing specs. That's where we started from, and it was good to have something concrete to discuss. But this is a brand new specification. Over time it has become its own thing."

Lilley said SVG's delays were not the fault of the wide array of technologies originally under consideration. The problem instead had to do with making SVG compatible with both established and emerging W3C recommendations for animation and other graphical effects.

For example, in the process of hammering out SVG, working group members decided to cook up a whole new way of handing animations with its key multimedia synchronization spec, SMIL Boston.

Synchronized Multimedia Integration Language (SMIL, pronounced "smile") Boston lets Web authors sync up sound, text and other multimedia elements using simple tags rather than programming code. SMIL animation, as reported in November, is designed to do the same thing for animations, turning the business of common Web animations such as mouse rollovers over to an XML dialect rather than to less-flexible scripting languages.

Proceeding in tandem with SVG, SMIL animation is nearing its own candidate recommendation, with a final working draft posted earlier this week.

The W3C stressed that SVG has been making progress in the marketplace even while hung up in the working group. Implementations exist in products by both IBM and Adobe Systems, among others.

As part of the candidate recommendation phase, the W3C is calling for more implementations of SVG; it has released a test suite that developers can use to evaluate their SVG work.

Copyright ©1995-2005 CNET Networks, Inc. All rights reserved.

NEWS.COM http://www.news.com/

W3C aims to streamline vector graphics

By Paul Festa
http://news.com.com/W3C+aims+to+streamline+vector+graphics/2100-1023_3-221604.html

Story last modified Thu Feb 11 17:15:00 PST 1999

An array of vector graphics proposals under consideration by the World Wide Web Consortium (W3C) are headed for consolidation under a new format.

The W3C today unveiled a working draft of Scalable Vector Graphics and solicited public comment on it. SVG follows four previous vector graphics submissions that the W3C has been considering, and is the first to be written in pure XML, or Extensible Markup Language.

Vector graphics are more flexible than the common bitmaps that form most of the graphics on the Web. In contrast to bitmaps, which are shipped fully rendered and are defined pixel by pixel, vector graphics are composed of mathematical descriptions of curves and forms. This composition results in a more compact file, the ability to render the image to fit different screens with varying resolutions, and greater ease in animating the image.

Out of the chaos of the four competing submissions, the W3C hopes to create order with SVG.

"We've been discussing how to create a new specification that has all the right features, significantly improves the state of graphics on the Web, and can be finished and implemented sooner rather than later," said Chris Lilley, W3C graphics activity lead and chair of the SVG working group. "The other specifications are interesting and useful starting points, but they are not under further development at W3C."

The four submissions that SVG supplants and gleans features from are as follows:

• Web Schematics, designed for making flow charts and other diagrams.

• Precision Graphics Markup Language, or PGML, which is best suited to graphics such as bar charts, logos, and screen graphics like push buttons. Submitted by Adobe, PGML is based on the firm's PostScript and portable document format (PDF) standards.

• Vector Markup Language, or VML, is a text format for vector graphics. A documentation of the internal graphics format in the yet-to-be-released Office 2000 suite from Microsoft, it integrates well with style sheets and facilitates the editing of images, according to backers.

• DrawML resembles Web Schematics, except that it relies heavily on Java classes to lay out diagrams.

These four submissions and SVG were designed to fix a host of problems with Web graphics,

Lilley said. The new format will make graphics searchable, compatible with style sheets, faster to download, better navigable by visually impaired users, and more faithfully printable.

SVG will be vendor-neutral and cross-platform, Lilley said. Written in XML, the format will be readable by any XML-enabled browser and compatible with the document object model (DOM).

SVG's proposed recommendation--the penultimate step in the W3C recommendation process--is slated for late summer.

Copyright ©1995-2005 CNET Networks, Inc. All rights reserved.

REC-smil-19980615

Synchronized Multimedia Integration Language (SMIL) 1.0 Specification

W3C Recommendation 15-June-1998

This version:
http://www.w3.org/TR/1998/REC-smil-19980615
Latest version:
http://www.w3.org/TR/REC-smil
Previous version:
http://www.w3.org/TR/1998/PR-smil-19980409

Copyright © 1998 W3C (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply. Your interactions with this site are in accordance with our public and Member privacy statements.

About this Document

This document has been prepared by the Synchronized Multimedia Working Group (WG) of the World Wide Web Consortium. The WG included the following individuals:

- Stephan Bugaj, Lucent/Bell Labs
- Dick Bulterman, CWI
- Bruce Butterfield, RealNetworks
- Wo Chang, NIST
- Guy Fouquet, Alcatel
- Christian Gran, GMD
- Mark Hakkinen, The Productivity Works
- Lynda Hardman, CWI
- Peter Hoddie, Apple
- Klaus Hofrichter, GMD
- Philipp Hoschka, W3C
- Jack Jansen, CWI
- George Kerscher, DAISY Consortium
- Rob Lanphier, RealNetworks
- Nabil Layaïda, INRIA
- Stephanie Leif, RealNetworks
- Sjoerd Mullender, CWI
- Didier Pillet, CNET/DSM
- Anup Rao, Netscape
- Lloyd Rutledge, CWI
- Patrick Soquet, Havas
- Warner ten Kate, Philips
- Jacco van Ossenbruggen, CWI
- Michael Vernick, Lucent/Bell Labs
- Jin Yu, DEC

Acknowledgments: In addition to the working group members, the following people contributed to the SMIL effort: Bert Bos (W3C), Dan Connolly (W3C), Patrick Deunhouwer (Philips), Martin Dürst (W3C), Al Gilman, Håkon Lie (W3C), Chris Lilley (W3C), Curtis Reynolds (RealNetworks), Michael Riesman, Curtis Reynolds (RealNetworks), Henning Schulzrinne (Columbia University) and Koga Youichirou (W3C).

Editor: Philipp Hoschka, W3C (hoschka@w3.org)

Abstract

This document specifies version 1 of the Synchronized Multimedia Integration Language (SMIL 1.0, pronounced "smile"). SMIL allows integrating a set of independent multimedia objects into a synchronized multimedia presentation. Using SMIL, an author can

1. describe the temporal behavior of the presentation
2. describe the layout of the presentation on a screen
3. associate hyperlinks with media objects

This specification is structured as follows: Section 1 presents the specification approach. Section 2 defines the "smil" element. Section 3 defines the elements that can be contained in the head part of a SMIL document. Section 4 defines the elements that can be contained in the body part of a SMIL document. In particular, this Section defines the time model used in SMIL. Section 5 describes the SMIL DTD.

Status of this Document

This document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

Comments on this Recommendation may be sent to the public mailing list www-smil@w3.org.

Available languages

The English version of this specification is the only normative version. However, for translations in other languages see http://www.w3.org/AudioVideo/SMIL/translations.

Errata

The list of known errors in this specification is available at http://www.w3.org/AudioVideo/SMIL/errata.

Table of Contents

1 Specification Approach

SMIL documents are XML 1.0 documents [XML10]. The reader is expected to be familiar with the concepts and terms defined in XML 1.0.

This specification does not rely on particular features defined in URLs that cannot potentially be expressed using URNs. Therefore, the more generic term URI [URI] is used throughout the specification.

The syntax of SMIL documents is defined by the DTD in Section 5.2. The syntax of an attribute value that cannot be defined using the DTD notation is defined together with the first element using an attribute that can contain the attribute value. The syntax of such attribute values is defined using the Extended Backus-Naur Form (EBNF) defined in the XML 1.0 specification.

An element definition is structured as follows: First, all attributes of the element are defined in alphabetical order. An attribute is defined in the following way: If the attribute is used by an element for the first time in the specification, the semantics of the attribute are defined. If the attribute has already been used by another element, the specification refers to the definition of the attribute in the first element that used it. The definition of element attributes is followed by the definition of any attribute values whose syntax cannot be defined using the DTD notation. The final section in an element

definition specifies the element content.

2 The `smil` Element

Element Attributes

The "smil" element can have the following attribute:

id
: This attribute uniquely identifies an element within a document. Its value is an XML identifier.

Element Content

The "smil" element can contain the following children:

body
: Defined in Section 4.1

head
: Defined in Section 3.1

3 The Document Head

3.1 The `head` Element

The "head" element contains information that is not related to the temporal behavior of the presentation.

Element Attributes

The "head" element can have the following attribute:

id
: Defined in Section 2

Element Content

The "head" element can contain the following children:

layout
: Defined in Section 3.2

meta
: Defined in Section 3.4

switch
: Defined in Section 4.3

The "head" element may contain any number of "meta" elements and either a "layout" element or a "switch" element.

3.2 The `layout` Element

The "layout" element determines how the elements in the document's body are positioned on an abstract rendering surface (either visual or acoustic).

If a document contains no layout element, the positioning of the body elements is implementation-dependent.

A SMIL document can contain multiple alternative layouts by enclosing several layout elements within a "switch" element (defined in Section 4.3). This can be used for example to describe the document's layout using different layout languages.

The following example shows how CSS2 can be used as alternative to the SMIL basic layout language (defined in Section 3.3):

```
<smil>
  <head>
    <switch>
     <layout type="text/css">
        [region="r"] { top: 20px; left: 20px }
     </layout>
     <layout>
       <region id="r" top="20" left="20" />
     </layout>
   </switch>
   </head>
   <body>
     <seq>
       <img region="r" src="http://www.w3.org/test" dur="10s" />
     </seq>
   </body>
</smil>
```

(note that in this example, both layout alternatives result in the same layout)

Element Attributes

id

Defined in Section 2

type

This attribute specifies which layout language is used in the layout element. If the player does not understand this language, it must skip all content up until the next "</layout>" tag. The default value of the type attribute is "text/smil-basic-layout".

Element Content

If the type attribute of the layout element has the value "text/smil-basic-layout", it can contain the following elements:

region

Defined in Section 3.3.1

root-layout

Defined in Section 3.3.2

If the type attribute of the "layout" element has another value, the element contains character data.

3.3 SMIL Basic Layout Language

This section defines a basic layout language for SMIL. SMIL basic layout is consistent with the visual rendering model defined in CSS2, it reuses the formatting properties defined by the CSS2 specification, and newly introduces the "fit" attribute [CSS2]. The reader is expected to be familiar with the concepts and terms defined in CSS2.

SMIL basic layout only controls the layout of media object elements (defined in Section 4.2.3). It is illegal to use SMIL basic layout for other SMIL elements.

The type identifier for SMIL basic layout is "text/smil-basic-layout".

Fixed Property Values

The following stylesheet defines the values of the CSS2 properties "display" and "position" that are valid in SMIL basic layout. These property values are fixed:

```
a           {display:block}
anchor      {display:block}
animation   {display: block;
             position: absolute}
body        {display: block}
head        {display: none}
img         {display: block;
             position: absolute}
layout      {display: none}
meta        {display: none}
par         {display: block}
region      {display: none}
ref         {display: block;
             position: absolute}
root-layout {display: none}
seq         {display: block}
smil        {display: block}
switch      {display:block}
text        {display: block;
             position: absolute}
textstream  {display: block;
             position: absolute}
video       {display: block;
             position: absolute}
```

Note that as a result of these definitions, all absolutely positioned elements (animation, img, ref, text, textstream and video) are contained within a single containing block defined by the content content edge of the root element (smil).

Default Values

SMIL basic layout defines default values for all layout-related attributes. These are consistent with the initial values of the corresponding properties in CSS2.

If the author wants to select the default layout values for *all* media object elements in a document, the document must contain an empty layout element of type "text/smil-basic-layout" such as:

```
<layout type="text/smil-basic-layout"></layout>
```

3.3.1 The `region` Element

The region element controls the position, size and scaling of media object elements.

In the following example fragment, the position of a text element is set to a 5 pixel distance from the top border of the rendering window:

```
<smil>
  <head>
    <layout>
      <region id="a" top="5" />
    </layout>
  </head>
  <body>
    <text region="a" src="text.html" dur="10s" />
  </body>
</smil>
```

Element Attributes

The "region" element can have the following attributes:

background-color
: The use and definition of this attribute are identical to the "background-color" property in the CSS2 specification, except that SMIL basic layout does not require support for "system colors". If the background-color attribute is absent, the background is transparent.

fit
: This attribute specifies the behavior if the intrinsic height and width of a visual media object differ from the values specified by the height and width attributes in the "region" element. This attribute does not have a 1-1 mapping onto a CSS2 property, but can be simulated in CSS2.
This attribute can have the following values:
  fill
  : Scale the object's height and width independently so that the content just touches all edges of the box.
  hidden
  : - If the intrinsic height (width) of the media object element is smaller than the height (width) defined in the "region" element, render the object starting from the top (left) edge and fill up the remaining height (width) with the background color.
    - If the intrinsic height (width) of the media object element is greater than the height (width) defined in the "region" element, render the object starting from the top (left) edge until the height (width) defined in the "region" element is reached, and clip the parts of the object below (right of) the height (width).
  meet
  : Scale the visual media object while preserving its aspect ratio until its height or width is equal to the value specified by the height or width attributes, while none of the content is clipped. The object's left top corner is positioned at the top-left coordinates of the box, and empty space at the left or bottom is filled up with the background color.
  scroll
  : A scrolling mechanism should be invoked when the element's rendered contents exceed its bounds.

slice
: Scale the visual media object while preserving its aspect ratio so that its height or width are equal to the value specified by the height and width attributes while some of the content may get clipped. Depending on the exact situation, either a horizontal or a vertical slice of the visual media object is displayed. Overflow width is clipped from the right of the media object. Overflow height is clipped from the bottom of the media object.

The default value of "fill" is "hidden".

height
: The use and definition of this attribute are identical to the "height" property in the CSS2 specification. Attribute values can be "percentage" values, and a variation of the "length" values defined in CSS2. For "length" values, SMIL basic layout only supports pixel units as defined in CSS2. It allows to leave out the "px" unit qualifier in pixel values (the "px" qualifier is required in CSS2).

id
: Defined in Section 2
A region element is applied to a positionable element by setting the region attribute of the positionable element to the id value of the region.
The "id" attribute is required for "region" elements.

left
: The use and definition of this attribute are identical to the "left" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.
The default value is zero.

skip-content
: This attribute is introduced for future extensibility of SMIL (see Appendix). It is interpreted in the following two cases:
  - If a new element is introduced in a future version of SMIL, and this element allows SMIL 1.0 elements as element content, the "skip-content" attribute controls whether this content is processed by a SMIL 1.0 player.
  - If an empty element in SMIL version 1.0 becomes non-empty in a future SMIL version, the "skip-content" attribute controls whether this content is ignored by a SMIL 1.0 player, or results in a syntax error.

  If the value of the "skip-content" attribute is "true", and one of the cases above apply, the content of the element is ignored. If the value is "false", the content of the element is processed.
  The default value for "skip-content" is "true".

title
: This attribute offers advisory information about the element for which it is set. Values of the title attribute may be rendered by user agents in a variety of ways. For instance, visual browsers frequently display the title as a "tool tip" (a short message that appears when the pointing device pauses over an object).
It is strongly recommended that all "region" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

top
: The use and definition of this attribute are identical to the "top" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.

The default value is zero.

width
The use and definition of this attribute are identical to the "width" property in the CSS2 specification. Attribute values have the same restrictions as the attribute values of the "height" attribute.

z-index
The use and definition of this attribute are identical to the "z-index" property in the CSS2 specification, with the following exception:

- If two boxes generated by elements A and B have the same stack level, then
  1. If the display of an element A starts later than the display of an element B, the box of A is stacked on top of the box of B (temporal order).
  2. If the display of the elements starts at the same time, and an element A occurs later in the SMIL document text than an element B, the box of A is stacked on top of the box of B (document tree order as defined in CSS2).

Element Content

"region" is an empty element.

3.3.2 The `root-layout` element

The "root-layout" element determines the value of the layout properties of the root element, which in turn determines the size of the viewport, e.g. the window in which the SMIL presentation is rendered.

If a document contains more than one "root-layout" element, this is an error, and the document should not be displayed.

Element Attributes

The "root-layout" element can have the following attributes:

background-color
Defined in Section 3.3.1

height
Defined in Section 3.3.1
Sets the height of the root element. Only length values are allowed.

id
Defined in Section 2

skip-content
Defined in Section 3.3.1

title
Defined in Section 3.3.1

width
Defined in Section 3.3.1
Sets the width of the root element. Only length values are allowed.

Element Content

"root-layout" is an empty element.

3.4 The meta Element

The "meta" element can be used to define properties of a document (e.g., author, expiration date, a list of key words, etc.) and assign values to those properties. Each "meta" element specifies a single property/value pair.

Element Attributes

The "meta" element can have the following attributes:

content
: This attribute specifies the value of the property defined in the meta element. The "content" attribute is required for "meta" elements.

id
: Defined in Section 2

name
: This attribute identifies the property defined in the meta element. The "name" attribute is required for "meta" elements.

skip-content
: Defined in Section 3.3.1

The list of properties is open-ended. This specification defines the following properties:

base
: The value of this property determines the base URI for all relative URIs used in the document.

pics-label or PICS-Label
: The value of this property specifies a valid rating label for the document as defined by PICS [PICS].

title
: The value of this property contains the title of the presentation.

Element Content

"meta" is an empty element.

4 The Document Body

4.1 The body Element

The "body" element contains information that is related to the temporal and linking behavior of the document. It implicitly defines a "seq" element (defined in Section 4.2.2, see Section 4.2.4 for a definition of the temporal semantics of the "body" element).

Element Attributes

The "body" element can have the following attribute:

id
: Defined in Section 2

Element Content

The "body" element can contain the following children:

a
Defined in Section 4.5.1
animation
Defined in Section 4.2.3
audio
Defined in Section 4.2.3
img
Defined in Section 4.2.3
par
Defined in Section 4.2.1
ref
Defined in Section 4.2.3
seq
Defined in Section 4.2.2
switch
Defined in Section 4.3
text
Defined in Section 4.2.3
textstream
Defined in Section 4.2.3
video
Defined in Section 4.2.3

4.2 Synchronization Elements

4.2.1 The `par` Element

The children of a par element can overlap in time. The textual order of appearance of children in a par has no significance for the timing of their presentation.

Element Attributes

The "par" element can have the following attributes:

abstract
A brief description of the content contained in the element.
author
The name of the author of the content contained in the element.
begin
This attribute specifies the time for the explicit begin of an element. See Section 4.2.4 for a definition of its semantics.
The attribute can contain the following two types of values:
delay-value
A delay value is a clock-value measuring presentation time. Presentation time advances at the speed of the presentation. It behaves like the timecode shown on a counter of a tape-deck. It can be stopped, decreased or increased either by user actions, or by the player itself.

The semantics of a delay value depend on the element's first ancestor that is a synchronization element (i.e. ancestors that are "a" or "switch" elements are ignored):

- If this ancestor is a "par" element, the value defines a delay from the effective begin of that element (see Figure 4.1).
- If this ancestor is a "seq" element (defined in Section 4.2.2), the value defines a delay from the effective end of the first lexical predecessor that is a synchronization element (see Figure 4.2).

event-value

The element begins when a certain event occurs (see Figure 4.3). Its value is an element-event (see Definition below).
The element generating the event must be "in scope". The set of "in scope" elements S is determined as follows:

1. Take all children from the element's first ancestor that is a synchronization element and add them to S.
2. Remove all "a" and "switch" elements from S. Add the children of all "a" elements to S, unless they are "switch" elements.

The resulting set S is the set of "in scope" elements.

```
<par>
  <audio id="a" begin="6s" src="audio" />
</par>
```

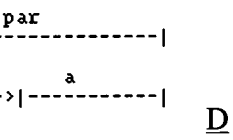


Figure 4.1: Using a delay value within a "par" element

```
<seq>
  <audio src="audio1" />
  <audio begin="5s" src="audio2" />
</seq>
```

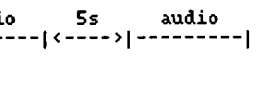
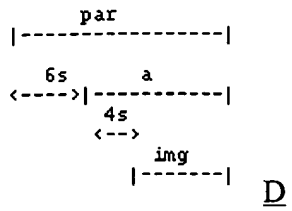


Figure 4.2: Using a delay value within a "seq" element

```
<par>
  <audio id="a" begin="6s" ... />
  <img  begin="id(a)(4s)" ... />
</par>
```



D

Figure 4.3: Synchronization attribute with element event value

copyright
: The copyright notice of the content contained in the element.

dur
: This attribute specifies the explicit duration of an element. See Section 4.2.4 for a definition of its semantics. The attribute value can be a clock value, or the string "indefinite".

end
: This attribute specifies the explicit end of an element. See Section 4.2.4 for a definition of its semantics. The attribute can contain the same types of attribute values as the "begin" attribute.

endsync
: For a definition of the semantics of this attribute, see Section 4.2.4. The attribute can have the following values:
  - first
    For a definition of the semantics of this value, see Section 4.2.4.
  - id-ref
    This attribute value has the following syntax:

    ```
    id-ref ::= "id(" id-value ")"
    ```

    where "id-value" must be a legal XML identifier.
    For a definition of the semantics of this value, see Section 4.2.4.
  - last
    For a definition of the semantics of this value, see Section 4.2.4.

  The default value of "endsync" is "last".

id
: Defined in Section 2

region
: This attribute specifies an abstract rendering surface (either visual or acoustic) defined within the layout section of the document. Its value must be an XML identifier. If no rendering surface with this id is defined in the layout section, the values of the formatting properties of this element are determined by the default layout.
  The "region" attribute on "par" elements cannot be used by the basic layout language for SMIL defined in this specification. It is added for completeness, since it may be required by other layout languages.

repeat
: For a definition of the semantics of this attribute, see Section 4.2.4. The attribute value can be an integer, or the string "indefinite". The default value is 1.

system-bitrate
: Defined in Section 4.4

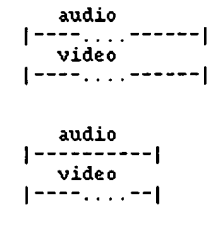
system-captions
: Defined in Section 4.4

system-language
Defined in Section 4.4
system-overdub-or-caption
Defined in Section 4.4
system-required
Defined in Section 4.4
system-screen-size
Defined in Section 4.4
system-screen-depth
Defined in Section 4.4
title
Defined in Section 3.3.1
It is strongly recommended that all "par" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

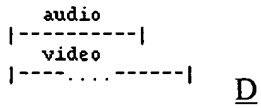
Note on Synchronization between Children

The accuracy of synchronization between the children in a parallel group is implementation-dependent. Take the example of synchronization in case of playback delays, i.e. the behavior when the "par" element contains two or more continuous media types such as audio or video, and one of them experiences a delay.
A player can show the following synchronization behaviors:

hard synchronization
The player synchronizes the children in the "par" element to a common clock (see Figure 4.4 a)).
soft synchronization
Each child of the "par" element has its own clock, which runs independently of the clocks of other children in the "par" element (see Figure 4.4 b)).



a) hard synchronization: Delay in video: Either the audio is stopped, or some video frames are dropped. The exact behavior is implementation-dependent



b) soft synchronization

Figure 4.4: Effect of a delay on playout schedule for players using different synchronization policies

Attribute Values

clock value

Clock values have the following syntax:

```
Clock-val             ::= Full-clock-val | Partial-clock-val | Timecount-val
Full-clock-val        ::= Hours ":" Minutes ":" Seconds ("." Fraction)?
Partial-clock-val     ::= Minutes ":" Seconds ("." Fraction)?
Timecount-val         ::= Timecount ("." Fraction)?
                          ("h" | "min" | "s" | "ms")? ; default is "s"
Hours                 ::= 2DIGIT; any positive number
Minutes               ::= 2DIGIT; range from 00 to 59
Seconds               ::= 2DIGIT; range from 00 to 59
Fraction              ::= DIGIT+
Timecount             ::= DIGIT+
2DIGIT                ::= DIGIT DIGIT
DIGIT                 ::= [0-9]
```

The following are examples of legal clock values:

- Full clock value: 02:30:03 = 2 hours, 30 minutes and 3 seconds
- Partial clock value: 02:33 = 2 minutes and 33 seconds
- Timecount values:
  3h = 3 hours
  45min = 45 minutes
  30s = 30 seconds
  5ms = 5 milliseconds

A fraction x with n digits represents the following value:

x * 1/10**n

Examples:

00.5s = 5 * 1/10 seconds = 500 milliseconds
00:00.005 = 5 * 1/1000 seconds = 5 milliseconds

element-event value

An *element event* value specifies a particular event in a synchronization element.
An element event has the following syntax:

```
Element-event     ::= "id(" Event-source ")(" Event ")"
Event-source      ::= Id-value
Event             ::= "begin" | Clock-val | "end"
```

The following events are defined:

begin

This event is generated at an element's effective begin.
Example use: `begin="id(x)(begin)"`

clock-val
: This event is generated when a clock associated with an element reaches a particular value. This clock starts at 0 at the element's effective begin. For "par" and "seq" elements, the clock gives the presentation time elapsed since the effective begin of the element. For media object elements, the semantics are implementation-dependent. The clock may either give presentation time elapsed since the effective begin, or it may give the media time of the object. The latter may differ from the presentation time that elapsed since the object's display was started e.g. due to rendering or network delays, and is the recommended approach.
It is an error to use a clock value that exceeds the value of the effective duration of the element generating the event.

Example use: `begin="id(x)(45s)"`

end
: This event is generated at the element's effective end.
Example use: `begin="id(x)(end)"`

Element Content

The par element can contain the following children:

a
: Defined in Section 4.5.1

animation
: Defined in Section 4.2.3

audio
: Defined in Section 4.2.3

img
: Defined in Section 4.2.3

par
: Defined in Section 4.2.1

ref
: Defined in Section 4.2.3

seq
: Defined in Section 4.2.2

switch
: Defined in Section 4.3

text
: Defined in Section 4.2.3

textstream
: Defined in Section 4.2.3

video
: Defined in Section 4.2.3

All of these elements may appear multiple times as direct children of a par element.

4.2.2 The `seq` Element

The children of a "seq" element form a temporal sequence.

Attributes

The seq element can have the following attributes:

abstract
Defined in Section 4.2.1
author
Defined in Section 4.2.1
begin
Defined in Section 4.2.1
copyright
Defined in Section 4.2.1
dur
Defined in Section 4.2.1
end
Defined in Section 4.2.1
id
Defined in Section 2
region
Defined in Section 4.2.1
The region attribute on "seq" elements cannot be used by the basic layout language for SMIL defined in this specification. It is added for completeness, since it may be required by other layout languages.
repeat
Defined in Section 4.2.1
system-bitrate
Defined in Section 4.4
system-captions
Defined in Section 4.4
system-language
Defined in Section 4.4
system-overdub-or-caption
Defined in Section 4.4
system-required
Defined in Section 4.4
system-screen-size
Defined in Section 4.4
system-screen-depth
Defined in Section 4.4
title
Defined in Section 3.3.1
It is strongly recommended that all "seq" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Element Content

The seq element can contain the following children:

a
Defined in Section 4.5.1

animation
Defined in Section 4.2.3
audio
Defined in Section 4.2.3
img
Defined in Section 4.2.3
par
Defined in Section 4.2.1
ref
Defined in Section 4.2.3
seq
Defined in Section 4.2.2
switch
Defined in Section 4.3
text
Defined in Section 4.2.3
textstream
Defined in Section 4.2.3
video
Defined in Section 4.2.3

4.2.3 Media Object Elements: The `ref`, `animation`, `audio`, `img`, `video`, `text` and `textstream` elements

The media object elements allow the inclusion of media objects into a SMIL presentation. Media objects are included by reference (using a URI).

There are two types of media objects: media objects with an intrinsic duration (e.g. video, audio) (also called "continuous media"), and media objects without intrinsic duration (e.g. text, image) (also called "discrete media").

Anchors and links can be attached to visual media objects, i.e. media objects rendered on a visual abstract rendering surface.

When playing back a media object, the player must not derive the exact type of the media object from the name of the media object element. Instead, it must rely solely on other sources about the type, such as type information contained in the "type" attribute, or the type information communicated by a server or the operating system.

Authors, however, should make sure that the group into which of the media object falls (animation, audio, img, video, text or textstream) is reflected in the element name. This is in order to increase the readability of the SMIL document. When in doubt about the group of a media object, authors should use the generic "ref" element.

Element Attributes

Media object elements can have the following attributes:

abstract
Defined in Section 4.2.1

alt

For user agents that cannot display a particular media-object, this attribute specifies alternate text. It is strongly recommended that all media object elements have an "alt" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

author

Defined in Section 4.2.1

begin

Defined in Section 4.2.1

clip-begin

The clip-begin attribute specifies the beginning of a sub-clip of a continuous media object as offset from the start of the media object.

Values in the clip-begin attribute have the following syntax:

```
Clip-time-value    ::= Metric "=" ( Clock-val | Smpte-val )
Metric             ::= Smpte-type | "npt"
Smpte-type         ::= "smpte" | "smpte-30-drop" | "smpte-25"
Smpte-val          ::= Hours ":" Minutes ":" Seconds
                       [ ":" Frames [ "." Subframes ]]
Hours              ::= 2DIGIT
Minutes            ::= 2DIGIT
Seconds            ::= 2DIGIT
Frames             ::= 2DIGIT
Subframes          ::= 2DIGIT
```

The value of this attribute consists of a metric specifier, followed by a time value whose syntax and semantics depend on the metric specifier. The following formats are allowed:

SMPTE Timestamp

SMPTE time codes [SMPTE] can be used for frame-level access accuracy. The metric specifier can have the following values:

smpte

smpte-30-drop

These values indicate the use of the "SMPTE 30 drop" format with 29.97 frames per second. The "frames" field in the time value can assume the values 0 through 29. The difference between 30 and 29.97 frames per second is handled by dropping the first two frame indices (values 00 and 01) of every minute, except every tenth minute.

smpte-25

The "frames" field in the time specification can assume the values 0 through 24.

The time value has the format hours:minutes:seconds:frames.subframes. If the frame value is zero, it may be omitted. Subframes are measured in one-hundredth of a frame.

Examples:

```
clip-begin="smpte=10:12:33:20"
```

Normal Play Time

Normal Play Time expresses time in terms of SMIL clock values. The metric specifier is "npt", and the syntax of the time value is identical to the syntax of SMIL clock values.

Examples:

```
clip-begin="npt=123.45s"
clip-begin="npt=12:05:35.3"
```

clip-end

The clip-end attribute specifies the end of a sub-clip of a continuous media object (such as audio,

video or another presentation) that should be played. It uses the same attribute value syntax as the clip-begin attribute.
If the value of the "clip-end" attribute exceeds the duration of the media object, the value is ignored, and the clip end is set equal to the effective end of the media object.

copyright
Defined in Section 4.2.1

dur
Defined in Section 4.2.1

end
Defined in Section 4.2.1

fill
For a definition of the semantics of this attribute, see Section 4.2.4. The attribute can have the values "remove" and "freeze".

id
Defined in Section 2

longdesc
This attribute specifies a link (URI) to a long description of a media object. This description should supplement the short description provided using the alt attribute. When the media-object has associated anchors, this attribute should provide information about the anchor's contents.

region
Defined in Section 4.2.1

src
The value of the src attribute is the URI of the media object.

system-bitrate
Defined in Section 4.4

system-captions
Defined in Section 4.4

system-language
Defined in Section 4.4

system-overdub-or-caption
Defined in Section 4.4

system-required
Defined in Section 4.4

system-screen-size
Defined in Section 4.4

system-screen-depth
Defined in Section 4.4

title
Defined in Section 3.3.1
It is strongly recommended that all media object elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

type
MIME type of the media object referenced by the "src" attribute.

Element Content

Media Object Elements can contain the following element:

anchor
Defined in Section 4.5.2

4.2.4 SMIL Time Model

4.2.4.1 Time Model Values

In the following discussion, the term "element" refers to synchronization elements only.

For each element we define the implicit, explicit, desired, and effective begin, duration, and end.

The effective begin/duration/end specify what the reader of the document will perceive.

The implicit, explicit, and desired values are auxiliary values used to define the effective values.

The rules for calculating each of these values for the elements defined in SMIL 1.0 are described in the next section.

1. Each element in SMIL has an *implicit begin*.
2. Each element can be assigned an *explicit begin* by adding a "begin" attribute to the element:

   ```
   begin = "value of explicit-begin"
   ```

   It is an error if the explicit begin is earlier than the implicit begin of the element.

3. Each element in SMIL has an *implicit end*.
4. Each element can be assigned an *explicit end* by adding an "end" attribute to the element:

   ```
   end = "value of explicit-end"
   ```

5. The *implicit duration* of an element is the difference between the implicit end and the implicit begin.
6. Each element in SMIL can be assigned an *explicit duration* by adding a "dur" attribute to the element:

   ```
   dur = "value of explicit-duration"
   ```

7. The *desired begin* of an element is equal to the explicit begin if one is given, otherwise the desired begin is equal to the implicit begin.
8. Each element has a *desired end*.
9. The *desired duration* of an element is the difference between the desired end and the desired begin.
10. Each element has an *effective begin*.
11. Each element has an *effective end*. (Note: the effective end of a child element can never be later than the effective end of its parent.)
12. The *effective duration* of an element is the difference between the effective end and the effective begin.

4.2.4.2 Determining Time Model Values for SMIL 1.0 Elements

This section defines how time model values are calculated for the synchronization elements of SMIL 1.0 in cases that are not covered by the rules in Section 4.2.4.1.

Determining the *implicit begin* of an element

- The implicit begin of the first child of the "body" element is when the document starts playing. When this is falls outside the scope of this document.
- The implicit begin of a child of a "par" element is equal to the effective begin of the "par" element.
- The implicit begin of the first child of a "seq" element is equal to the effective begin of the "seq" element.
- The implicit begin of any other child of a "seq" element is equal to the desired end time of the previous child of the "seq" element.

Determining the *implicit end* of an element

The first description that matches the element is the one that is to be used:

- An element with a "repeat" attribute with value "indefinite" has an implicit end immediately after its effective begin.
- An element with a "repeat" attribute with a value other than "indefinite" has an implicit end equal to the implicit end of a seq element with the stated number of copies of the element without "repeat" attribute as children.
- A media object element referring to a continuous media object has an implicit end equal to the sum of the effective begin of the element and the intrinsic duration of the media object.
- A media object element referring to a discrete media object such as text or image has an implicit end immediately after its effective begin.
- A "seq" element has an implicit end equal to the desired end of its last child.
- A "par" element has an implicit end that depends on the value of the "endsync" attribute. The implicit end is equal to the sum of the effective begin of the "par" element and the implicit duration which is derived as follows:
    - If the value of the "endsync" attribute is "last", or if the "endsync" attribute is missing, the implicit duration of the "par" element is the maximum of the desired durations of its children.
    - If the value of the "endsync" attribute is "first", the implicit duration of the "par" element is the minimum of the desired durations of its children.
    - If the value of the "endsync" attribute is an id-ref, the implicit duration of the "par" element is equal to the desired duration of the child referenced by the "id-ref".

Determining the *desired end* of an element

- If the element has both an explicit duration and an explicit end, the desired end is the minimum of:
    - the sum of the desired begin and the explicit duration; and
    - the explicit end.
- If the element has an explicit duration but no explicit end, the desired end is the sum of the desired begin and the explicit duration.
- If the element has an explicit end but no explicit duration, the desired end is equal to the explicit end
- Otherwise, the desired end is equal to the implicit end.

Determining the *desired begin* of an element

The desired begin of an element is determined by using rule 7 in Section 4.2.4.1.

Determining the *effective begin* of an element

The *effective begin* of an element is equal to the desired begin of the element, unless the effective end of the parent element is earlier than this time, in which case the element is not shown at all.

Determining the *effective end* of an element

- The effective end of the last child of the body element is player-dependent. The effective end is at least as late as the desired end, but whether it is any later is implementation-dependent.
- The effective end of the child of a "par" element can be derived as follows:
  - If the child has a "fill" attribute, and the value of the "fill" attribute is "freeze", the effective end of the child element is equal to the effective end of the parent.
    The last state of the element is retained on the screen until the effective end of the element.
  - If the child has a "fill" attribute, and the value of the "fill" attribute is "remove", the effective end of the child element is the minimum of the effective end of the parent and the desired end of the child element.
  - If the child element has no "fill" attribute, the effective end of the child depends on whether or not the child has an explicit duration or end.
    - If the child has an explicit duration or end, the effective end is determined as if the element had a "fill" attribute with value "remove".
    - If the child has neither an explicit duration nor an explicit end, the effective end is determined as if the element had a "fill" attribute with value "freeze".
- The effective end of the last child of a "seq" element is derived in the same way as the effective end of a child of a "par" element.
- The effective end of any other child of a "seq" element can be derived as follows:
  - If the child has a "fill" attribute, and the value of the "fill" attribute is "freeze", the effective end of the child element is equal to the effective begin of the next element
  - If the child has a "fill" attribute, and the value of the "fill" attribute is "remove", the effective end of the child element is the minimum of the effective begin of the next element and the desired end of the next child element.
  - If the child element has no "fill" attribute, the effective end of the child depends on whether or not the child has an explicit duration or end.
    - If the child has an explicit duration or end, the effective end is determined as if the element had a fill attribute with value "remove".
    - If the child has neither an explicit duration nor an explicit end, the effective end is determined as if the element had a fill attribute with value "freeze".

4.3 The `switch` Element

The switch element allows an author to specify a set of alternative elements from which only one acceptable element should be chosen. An element is acceptable if the element is a SMIL 1.0 element, the media-type can be decoded, and all of the test-attributes (see Section 4.4) of the element evaluate to "true".

An element is selected as follows: the player evaluates the elements in the order in which they occur in the switch element. The first acceptable element is selected at the exclusion of all other elements within the switch.

Thus, authors should order the alternatives from the most desirable to the least desirable. Furthermore, authors should place a relatively fail-safe alternative as the last item in the <switch> so that at least one

item within the switch is chosen (unless this is explicitly not desired). Implementations should NOT arbitrarily pick an object within a <switch> when test-attributes for all fail.

Note that http URIs provide for content-negotiation, which may be an alternative to using the "switch" element in some cases.

Attributes

The switch element can have the following attributes:

id
Defined in Section 2
title
Defined in Section 3.3.1
It is strongly recommended that all switch elements have a "title" attribute with a meaningful description Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Element Content

If the "switch" element is used as a direct or indirect child of a "body" element, it can contain the following children:

a
Defined in Section 4.5.1
animation
Defined in Section 4.2.3
audio
Defined in Section 4.2.3
img
Defined in Section 4.2.3
par
Defined in Section 4.2.1
ref
Defined in Section 4.2.3
seq
Defined in Section 4.2.2
switch
Defined in Section 4.3
text
Defined in Section 4.2.3
textstream
Defined in Section 4.2.3
video
Defined in Section 4.2.3

All of these elements may appear multiple times as children of a "switch" element.

If the "switch" element is used within a "head" element, it can contain the following child:

layout
Defined in Section 3.2
Multiple layout elements may occur within the switch element.

4.4 Test Attributes

This specification defines a list of test attributes that can be added to any synchronization element, and that test system capabilities and settings. Conceptually, these attributes represent boolean tests. When one of the test attributes specified for an element evaluates to "false", the element carrying this attribute is ignored.

Within the list below, the concept of "user preference" may show up. User preferences are usually set by the playback engine using a preferences dialog box, but this specification does not place any restrictions on how such preferences are communicated from the user to the SMIL player.

The following test attributes are defined in SMIL 1.0:

system-bitrate
This attribute specifies the approximate bandwidth, in bits per second available to the system. The measurement of bandwidth is application specific, meaning that applications may use sophisticated measurement of end-to-end connectivity, or a simple static setting controlled by the user. In the latter case, this could for instance be used to make a choice based on the users connection to the network. Typical values for modem users would be 14400, 28800, 56000 bit/s etc. Evaluates to "true" if the available system bitrate is equal to or greater than the given value. Evaluates to "false" if the available system bitrate is less than the given value.
The attribute can assume any integer value greater than 0. If the value exceeds an implementation-defined maximum bandwidth value, the attribute always evaluates to "false".

system-captions
This attribute allows authors to distinguish between a redundant text equivalent of the audio portion of the presentation (intended for a audiences such as those with hearing disabilities or those learning to read who want or need this information) and text intended for a wide audience. The attribute can has the value "on" if the user has indicated a desire to see closed-captioning information, and it has the value "off" if the user has indicated that they don't wish to see such information. Evaluates to "true" if the value is "on", and evaluates to "false" if the value is "off".

system-language
The attribute value is a comma-separated list of language names as defined in [RFC1766].

Evaluates to "true" if one of the languages indicated by user preferences exactly equals one of the languages given in the value of this parameter, or if one of the languages indicated by

user preferences exactly equals a prefix of one of the languages given in the value of this parameter such that the first tag character following the prefix is "-".

Evaluates to "false" otherwise.

Note: This use of a prefix matching rule does not imply that language tags are assigned to languages in such a way that it is always true that if a user understands a language with a certain tag, then this user will also understand all languages with tags for which this tag is a prefix.

The prefix rule simply allows the use of prefix tags if this is the case.

Implementation note: When making the choice of linguistic preference available to the user, implementors should take into account the fact that users are not familiar with the details of language matching as described above, and should provide appropriate guidance. As an example, users may assume that on selecting "en-gb", they will be served any kind of English document if British English is not available. The user interface for setting user preferences should guide the user to add "en" to get the best matching behavior.

Multiple languages MAY be listed for content that is intended for multiple audiences. For example, a rendition of the "Treaty of Waitangi", presented simultaneously in the original Maori and English versions, would call for:

```
<audio src="foo.rm" system-language="mi, en"/>
```

However, just because multiple languages are present within the object on which the system-language test attribute is placed, this does not mean that it is intended for multiple linguistic audiences. An example would be a beginner's language primer, such as "A First Lesson in Latin," which is clearly intended to be used by an English-literate audience. In this case, the system-language test attribute should only include "en".

Authoring note: Authors should realize that if several alternative language objects are enclosed in a "switch", and none of them matches, this may lead to situations such as a video being shown without any audio track. It is thus recommended to include a "catch-all" choice at the end of such a switch which is acceptable in all cases.

system-overdub-or-caption
: This attribute is a setting which determines if users prefer overdubbing or captioning when the option is available. The attribute can have the values "caption" and "overdub". Evaluates to "true" if the user preference matches this attribute value. Evaluates to "false" if they do not match.

system-required
: This attribute specifies the name of an extension. Evaluates to "true" if the extension is supported by the implementation, otherwise, this evaluates to "false". In a future version of SMIL, this attribute value will be an XML namespace [NAMESPACES].

system-screen-size
: Attribute values have the following syntax:
`screen-size-val ::= screen-height"X"screen-width`
Each of these is a pixel value, and must be an integer value greater than 0. Evaluates to "true" if the SMIL playback engine is capable of displaying a presentation of the given size. Evaluates to "false" if the SMIL playback engine is only capable of displaying a smaller presentation.

system-screen-depth
: This attribute specifies the depth of the screen color palette in bits required for displaying the element. The value must be greater than 0. Typical values are 1, 8, 24 .... Evaluates to "true" if the SMIL playback engine is capable of displaying images or video with the given color depth. Evaluates to "false" if the SMIL playback engine is only capable of displaying images or video with a smaller color depth.

Examples

1) Choosing between content with different bitrate

In a common scenario, implementations may wish to allow for selection via a "system-bitrate" parameter on elements. The media player evaluates each of the "choices" (elements within the switch)

one at a time, looking for an acceptable bitrate given the known characteristics of the link between the media player and media server.

```
...
<par>
  <text .../>
  <switch>
    <par system-bitrate="40000">
    ...
    </par>
    <par system-bitrate="24000">
    ...
    </par>
    <par system-bitrate="10000">
    ........
    </par>
  </switch>
</par>
...
```

2) Choosing between audio resources with different bitrate

The elements within the switch may be any combination of elements. For instance, one could merely be specifying an alternate audio track:

```
...
<switch>
   <audio src="joe-audio-better-quality" system-bitrate="16000" />
   <audio src="joe-audio" system-bitrate="8000" />
</switch>
...
```

3) Choosing between audio resources in different languages

In the following example, an audio resource is available both in French and in English. Based on the user's preferred language, the player can choose one of these audio resources.

```
...
<switch>
   <audio src="joe-audio-french" system-language="fr"/>
   <audio src="joe-audio-english" system-language="en"/>
</switch>
...
```

4) Choosing between content written for different screens

In the following example, the presentation contains alternative parts designed for screens with different resolutions and bit-depths. Depending on the particular characteristics of the screen, the player can choose one of the alternatives.

```
...
<par>
  <text .../>
  <switch>
    <par system-screen-size="1280X1024" system-screen-depth="16">
```

```
    ........
    </par>
    <par system-screen-size="640X480" system-screen-depth="32">
    ...
    </par>
    <par system-screen-size="640X480" system-screen-depth="16">
    ...
    </par>
  </switch>
</par>
...
```

5) Distinguishing caption tracks from stock tickers

In the following example, captions are shown only if the user wants captions on.

```
...
<seq>
  <par>
    <audio      src="audio.rm"/>
    <video      src="video.rm"/>
    <textstream src="stockticker.rtx"/>
    <textstream src="closed-caps.rtx" system-captions="on"/>
  </par>
</seq>
...
```

6) Choosing the language of overdub and caption tracks

In the following example, a French-language movie is available with English, German, and Dutch overdub and caption tracks. The following SMIL segment expresses this, and switches on the alternatives that the user prefers.

```
...
<par>
  <switch>
    <audio src="movie-aud-en.rm" system-language="en"
                system-overdub-or-caption="overdub"/>
    <audio src="movie-aud-de.rm" system-language="de"
                system-overdub-or-caption="overdub"/>
    <audio src="movie-aud-nl.rm" system-language="nl"
                system-overdub-or-caption="overdub"/>
     <!-- French for everyone else -->
     <audio src="movie-aud-fr.rm"/>
   </switch>
   <video src="movie-vid.rm"/>
   <switch>
     <textstream src="movie-caps-en.rtx" system-language="en"
                 system-overdub-or-caption="caption"/>
     <textstream src="movie-caps-de.rtx" system-language="de"
                system-overdub-or-caption="caption"/>
     <textstream src="movie-caps-nl.rtx" system-language="nl"
                 system-overdub-or-caption="caption"/>
     <!-- French captions for those that really want them -->
     <textstream src="movie-caps-fr.rtx" system-captions="on"/>
   </switch>
</par>
```

. . .

4.5 Hyperlinking Elements

The link elements allows the description of navigational links between objects.

SMIL provides only for in-line link elements. Links are limited to uni-directional single-headed links (i.e. all links have exactly one source and one destination resource). All links in SMIL are actuated by the user.

Handling of Links in Embedded Documents

Due to its integrating nature, the presentation of a SMIL document may involve other (non-SMIL) applications or plug-ins. For example, a SMIL browser may use an HTML plug-in to display an embedded HTML page. Vice versa, an HTML browser may use a SMIL plug-in to display a SMIL document embedded in an HTML page.

In such presentations, links may be defined by documents at different levels and conflicts may arise. In this case, the link defined by the containing document should take precedence over the link defined by the embedded object. Note that since this might require communication between the browser and the plug-in, SMIL implementations may choose not to comply with this recommendation.

If a link is defined in an embedded SMIL document, traversal of the link affects only the embedded SMIL document.

If a link is defined in a non-SMIL document which is embedded in a SMIL document, link traversal can only affect the presentation of the embedded document and not the presentation of the containing SMIL document. This restriction may be released in future versions of SMIL.

Addressing

SMIL supports name fragment identifiers and the '#' connector. This means that SMIL supports locators as currently used in HTML (e.g. it uses locators of the form "http://foo.com/some/path#anchor1").

Linking to SMIL Fragments

A locator that points to a SMIL document may contain a fragment part (e.g. http://www.w3.org/test.smi#par1). The fragment part is an id value that identifies one of the elements within the referenced SMIL document. If a link containing a fragment part is followed, the presentation should start as if the user had fast-forwarded the presentation represented by the destination document to the effective begin of the element designated by the fragment.

The following special cases can occur:

1. The element addressed by the link has a "repeat" attribute.
    1. If the value of the "repeat" attribute is N, all N repetitions of the element are played.
    2. If the value of the "repeat" attribute is "indefinite", playback ends according to the rules defined for repeat value "indefinite".
2. The element addressed by the link is contained within another element that contains a "repeat" attribute.

1. If the value of the "repeat" attribute is N, playback starts at the beginning of the element addressed by the link, followed by N-1 repetitions of the element containing the "repeat" attribute.
2. If the value of the "repeat" attribute is "indefinite", playback starts at the beginning of the element addressed by the link. Playback ends according to the rules defined for repeat value "indefinite".

3. The element addressed by the link is content of a "switch" element: It is forbidden to link to elements that are the content of "switch" elements.

4.5.1 The a Element

The functionality of the "a" element is very similar to the functionality of the "a" element in HTML 4.0 [HTML40] . SMIL adds an attribute "show" that controls the temporal behavior of the source when the link is followed. For synchronization purposes, the "a" element is transparent, i.e. it does not influence the synchronization of its child elements. "a" elements may not be nested. An "a" element must have an href attribute.

Attributes

The "a" element can have the following attributes:

id
: Defined in Section 2

href
: This attribute contains the URI of the link's destination.
The "href" attribute is required for "a" elements.

show
: This attribute controls the behavior of the source document containing the link when the link is followed. It can have one of the following values:
  - "replace": The current presentation is paused at its current state and is replaced by the destination resource. If the player offers a history mechanism, the source presentation resumes from the state in which it was paused when the user returns to it.
  - "new": The presentation of the destination resource starts in a new context, not affecting the source resource.
  - "pause": The source presentation is paused at its current state, and the destination resource starts in a new context. When the display of the destination resource ends, the source presentation resumes from the state in which it was paused.

  The default value of "show" is "replace".

title
: Defined in Section 3.3.1
It is strongly recommended that all "a" elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Element Content

The "a" element can contain the following children:

animation
Defined in Section 4.2.3
audio
Defined in Section 4.2.3
img
Defined in Section 4.2.3
par
Defined in Section 4.2.1
ref
Defined in Section 4.2.3
seq
Defined in Section 4.2.2
switch
Defined in Section 4.3
text
Defined in Section 4.2.3
textstream
Defined in Section 4.2.3
video
Defined in Section 4.2.3

Examples

Example 1

The link starts up the new presentation replacing the presentation that was playing.

```
<a href="http://www.cwi.nl/somewhereelse.smi">
     <video src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

In the example, the second line can be replaced by a reference to any valid subtree of an SMIL presentation.

Example 2

The link starts up the new presentation in addition to the presentation that was playing.

```
<a href="http://www.cwi.nl/somewhereelse.smi" show="new">
     <video src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

For example, this allows a SMIL player to spawn off an HTML browser.

Example 3

The link starts up the new presentation and pauses the presentation that was playing.

```
<a href="http://www.cwi.nl/somewhereelse.smi" show="pause">
     <video src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>
```

Example 4

The following example contains a link from an element in one presentation A to the middle of another presentation B. This would play presentation B starting from the effective begin of the element with id "next".

```
Presentation A:

     <a href="http://www.cwi.nl/presentationB#next">
       <video src="rtsp://foo.com/graph.imf"/>
     </a>


Presentation B (http://www.cwi.nl/presentation):

      ...
      <seq>
        <video src="rtsp://foo.com/graph.imf"/>
        <par>
          <video src="rtsp://foo.com/timbl.rm" region="l_window"/>
          <video id="next" src="rtsp://foo.com/v1.rm" region="r_window"/>
                 ^^^^^^^^^
          <text src="rtsp://foo.com/caption1.html" region="l_2_title"/>
          <text src="rtsp://foo.com/caption2.rtx" region="r_2_title"/>
        </par>
      </seq>
      ...
```

4.5.2 The anchor Element

The functionality of the "a" element is restricted in that it only allows associating a link with a complete media object. HTML image maps have demonstrated that it is useful to associate links with spatial subparts of an object. The anchor element realizes similar functionality for SMIL:

1. The anchor element allows associating a link destination to spatial and temporal subparts of a media object, using the "href" attribute (in contrast, the "a" element only allows associating a link with a complete media object).
2. The anchor element allows making a subpart of the media object the destination of a link, using the "id" attribute.
3. The anchor element allows breaking up an object into spatial subparts, using the "coords" attribute.
4. The anchor element allows breaking up an object into temporal subparts, using the "begin" and "end" attributes. The values of the begin and end attributes are relative to the beginning of the media object.

Attributes

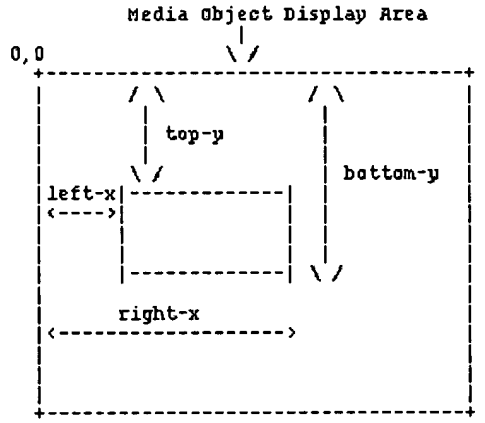
The anchor element can have the following attributes:

begin
: Defined in Section 4.2.1

coords
: The value of this attribute specifies a rectangle within the display area of a visual media object.

Syntax and semantics of this attribute are similar to the coords attribute in HTML image maps, when the link is associated with a rectangular shape. The rectangle is specified by four length values: The first two values specify the coordinates of the upper left corner of the rectangle.The second two values specify the coordinates of the lower right corner of the rectangle. Coordinates are relative to the top left corner of the visual media object (see Figure 4.5). If a coordinate is specified as a percentage value, it is relative to the total width or height of the media object display area.
An attribute with an erroneous coords value is ignored (right-x smaller or equal to left-x, bottom-y smaller or equal to top-y). If the rectangle defined by the coords attribute exceeds the area covered by the media object, exceeding height and width are clipped at the borders of the media object.
Values of the coords attribute have the following syntax:

```
coords-value ::= left-x "," top-y "," right-x "," bottom-y
```



D

Figure 4.5: Semantics of "coords" attribute

end
Defined in Section 4.2.1
id
Defined in Section 2
show
Defined in Section 4.5.1
skip-content
Defined in Section 3.3.1
title
Defined in Section 3.3.1
It is strongly recommended that all anchor elements have a "title" attribute with a meaningful description. Authoring tools should ensure that no element can be introduced into a SMIL document without this attribute.

Examples

1) Associating links with spatial subparts

In the following example, the screenspace taken up by a video clip is split into two sections. A different link is associated with each of these sections.

```
<video src="http://www.w3.org/CoolStuff">
  <anchor href="http://www.w3.org/AudioVideo" coords="0%,0%,50%,50%"/>
  <anchor href="http://www.w3.org/Style"      coords="50%,50%,100%,100%"/>
</video>
```

2) Associating links with temporal subparts

In the following example, the duration of a video clip is split into two subintervals. A different link is associated with each of these subintervals.

```
<video src="http://www.w3.org/CoolStuff">
  <anchor href="http://www.w3.org/AudioVideo" begin="0s" end="5s"/>
  <anchor href="http://www.w3.org/Style"      begin="5s" end="10s"/>
</video>
```

3) Jumping to a subpart of an object

The following example contains a link from an element in one presentation A to the middle of a video object contained in another presentation B. This would play presentation B starting from second 5 in the video (i.e. the presentation would start as if the user had fast-forwarded the whole presentation to the point at which the designated fragment in the "CoolStuff" video begins).

```
Presentation A:

<a href="http://www.cwi.nl/mm/presentationB#tim">
   <video id="graph" src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>


Presentation B:

<video src="http://www.w3.org/CoolStuff">
  <anchor id="joe" begin="0s" end="5s"/>
  <anchor id="tim" begin="5s" end="10s"/>
</video>
```

4) Combining different uses of links

The following example shows how the different uses of associated links can be used in combination.

```
Presentation A:

<a href="http://www.cwi.nl/mm/presentationB#tim">
  <video id="graph" src="rtsp://foo.com/graph.imf" region="l_window"/>
</a>


Presentation B:

<video src="http://www.w3.org/CoolStuff">
  <anchor id="joe" begin="0s" end="5s" coords="0%,0%,50%,50%"
          href="http://www.w3.org/"/>
```

```
  <anchor id="tim" begin="5s" end="10s" coords="0%,0%,50%,50%"
          href="http://www.w3.org/Tim"/>
</video>
```

5 SMIL DTD

5.1 Relation to XML

A SMIL 1.0 document may optionally contain a document type declaration, which names the document type definition (DTD) in use for the document. For SMIL, the document type declaration should look as follows (the double quotes can be replaced by single quotes):

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
                "http://www.w3.org/TR/REC-smil/SMIL10.dtd">
```

The XML 1.0 specification provides a way to extend the DTD using the <!DOCTYPE> element, for instance to add a new set of entity definitions. Authors must not use this feature with SMIL since many SMIL players will not support it.

The following is illegal in SMIL:

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
                "http://www.w3.org/TR/REC-smil/SMIL10.dtd" [
<!ENTITY % AcmeCorpSymbols PUBLIC
     "-//Acme Corp//ENTITIES Corporate Symbols//EN"
     "http://www.acme.com/corp_symbols.xml"
>
%AcmeCorpSymbols;
]>
```

5.2 DTD

```
<!--

    This is the XML document type definition (DTD) for SMIL 1.0.

    Date: 1998/06/15 08:56:30

    Authors:
        Jacco van Ossenbruggen <jrvosse@cwi.nl>
        Sjoerd Mullender       <sjoerd@cwi.nl>

    Further information about SMIL is available at:

          http://www.w3.org/AudioVideo/

-->

<!-- Generally useful entities -->
<!ENTITY % id-attr "id ID #IMPLIED">
<!ENTITY % title-attr "title CDATA #IMPLIED">
<!ENTITY % skip-attr "skip-content (true|false) 'true'">
<!ENTITY % desc-attr "
        %title-attr;
        abstract        CDATA   #IMPLIED
```

```
        author          CDATA   #IMPLIED
        copyright       CDATA   #IMPLIED
">

<!--==================== SMIL Document ======================================-->
<!--
     The root element SMIL contains all other elements.
-->
<!ELEMENT smil (head?,body?)>
<!ATTLIST smil
        %id-attr;
>

<!--==================== The Document Head ==================================-->
<!ENTITY % layout-section "layout|switch">

<!ENTITY % head-element "(meta*,((%layout-section;), meta*))?">

<!ELEMENT head %head-element;>
<!ATTLIST head %id-attr;>


<!--==================== Layout Element =====================================-->
<!--
     Layout contains the region and root-layout elements defined by
     smil-basic-layout or other elements defined an external layout
     mechanism.
-->
<!ELEMENT layout ANY>
<!ATTLIST layout
        %id-attr;
        type CDATA      "text/smil-basic-layout"
>


<!--==================== Region Element =====================================-->
<!ENTITY % viewport-attrs "
        height              CDATA    #IMPLIED
        width               CDATA    #IMPLIED
        background-color    CDATA    #IMPLIED
">

<!ELEMENT region EMPTY>
<!ATTLIST region
        %id-attr;
        %title-attr;
        %viewport-attrs;
        left                CDATA    "0"
        top                 CDATA    "0"
        z-index             CDATA    "0"
        fit                 (hidden|fill|meet|scroll|slice)    "hidden"
        %skip-attr;
>

<!--==================== Root-layout Element ================================-->
<!ELEMENT root-layout EMPTY>
<!ATTLIST root-layout
        %id-attr;
        %title-attr;
        %viewport-attrs;
```

```
        %skip-attr;
>


<!--==================== Meta Element=======================================-->
<!ELEMENT meta EMPTY>
<!ATTLIST meta
        name    NMTOKEN #REQUIRED
        content CDATA   #REQUIRED
        %skip-attr;
>

<!--==================== The Document Body ==================================-->
<!ENTITY % media-object "audio|video|text|img|animation|textstream|ref">
<!ENTITY % schedule "par|seq|(%media-object;)">
<!ENTITY % inline-link "a">
<!ENTITY % assoc-link "anchor">
<!ENTITY % link "%inline-link;">
<!ENTITY % container-content "(%schedule;)|switch|(%link;)">
<!ENTITY % body-content "(%container-content;)">

<!ELEMENT body (%body-content;)*>
<!ATTLIST body %id-attr;>

<!--==================== Synchronization Attributes =========================-->
<!ENTITY % sync-attributes "
        begin   CDATA   #IMPLIED
        end     CDATA   #IMPLIED
">

<!--==================== Switch Parameter Attributes ========================-->
<!ENTITY % system-attribute "
        system-bitrate              CDATA               #IMPLIED
        system-language             CDATA               #IMPLIED
        system-required             NMTOKEN             #IMPLIED
        system-screen-size          CDATA               #IMPLIED
        system-screen-depth         CDATA               #IMPLIED
        system-captions             (on|off)            #IMPLIED
        system-overdub-or-caption   (caption|overdub)   #IMPLIED
">

<!--==================== Fill Attribute =====================================-->
<!ENTITY % fill-attribute "
        fill    (remove|freeze)    'remove'
">


<!--==================== The Parallel Element ===============================-->
<!ENTITY % par-content "%container-content;">
<!ELEMENT par    (%par-content;)*>
<!ATTLIST par
        %id-attr;
        %desc-attr;
        endsync CDATA           "last"
        dur     CDATA           #IMPLIED
        repeat  CDATA           "1"
        region  IDREF           #IMPLIED
        %sync-attributes;
        %system-attribute;
>
```

```
<!--==================== The Sequential Element ============================-->
<!ENTITY % seq-content "%container-content;">
<!ELEMENT seq    (%seq-content;)*>
<!ATTLIST seq
        %id-attr;
        %desc-attr;
        dur     CDATA           #IMPLIED
        repeat  CDATA           "1"
        region  IDREF           #IMPLIED
        %sync-attributes;
        %system-attribute;
>

<!--==================== The Switch Element ================================-->
<!-- In the head, a switch may contain only layout elements,
     in the body, only container elements. However, this
     constraint cannot be expressed in the DTD (?), so
     we allow both:
-->
<!ENTITY % switch-content "layout|(%container-content;)">
<!ELEMENT switch (%switch-content;)*>
<!ATTLIST switch
        %id-attr;
        %title-attr;
>

<!--==================== Media Object Elements =============================-->
<!-- SMIL only defines the structure. The real media data is
     referenced by the src attribute of the media objects.
-->

<!-- Furthermore, they have the following attributes as defined
     in the SMIL specification:
-->
<!ENTITY % mo-attributes "
        %id-attr;
        %desc-attr;
        region     IDREF          #IMPLIED
        alt        CDATA          #IMPLIED
        longdesc   CDATA          #IMPLIED
        src        CDATA          #IMPLIED
        type       CDATA          #IMPLIED
        dur        CDATA          #IMPLIED
        repeat     CDATA          '1'
        %fill-attribute;
        %sync-attributes;
        %system-attribute;
">

<!--
     Most info is in the attributes, media objects are empty or
     contain associated link elements:
-->
<!ENTITY % mo-content "(%assoc-link;)*">
<!ENTITY % clip-attrs "
        clip-begin      CDATA   #IMPLIED
        clip-end        CDATA   #IMPLIED
">
```

```
<!ELEMENT ref           %mo-content;>
<!ELEMENT audio         %mo-content;>
<!ELEMENT img           %mo-content;>
<!ELEMENT video         %mo-content;>
<!ELEMENT text          %mo-content;>
<!ELEMENT textstream    %mo-content;>
<!ELEMENT animation     %mo-content;>

<!ATTLIST ref           %mo-attributes; %clip-attrs;>
<!ATTLIST audio         %mo-attributes; %clip-attrs;>
<!ATTLIST video         %mo-attributes; %clip-attrs;>
<!ATTLIST animation     %mo-attributes; %clip-attrs;>
<!ATTLIST textstream    %mo-attributes; %clip-attrs;>
<!ATTLIST text          %mo-attributes;>
<!ATTLIST img           %mo-attributes;>

<!--==================== Link Elements ====================================-->

<!ENTITY % smil-link-attributes "
        %id-attr;
        %title-attr;
        href            CDATA                   #REQUIRED
        show            (replace|new|pause)     'replace'
">


<!--==================== Inline Link Element ==============================-->
<!ELEMENT a (%schedule;|switch)*>
<!ATTLIST a
        %smil-link-attributes;
>


<!--==================== Associated Link Element ==========================-->
<!ELEMENT anchor EMPTY>
<!ATTLIST anchor
        %skip-attr;
        %smil-link-attributes;
        %sync-attributes;
        coords          CDATA                   #IMPLIED
>
```

References

[CSS2]
"Cascading Style Sheets, level 2", B. Bos, H. Lie, C. Lilley, I. Jacobs, 12 May 1998. Available at http://www.w3.org/TR/REC-CSS2/.

[HTML40]
"HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, 24 April 1998. Available at http://www.w3.org/TR/REC-html40/.

[ISO/IEC 10646]
ISO (International Organization for Standardization). ISO/IEC 10646-1993 (E). Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. [Geneva]: International Organization for Standardization, 1993 (plus amendments AM 1 through AM 7).

[NAMESPACES]

"Namespaces in XML", T. Bray, D. Hollander, A. Layman, 27 March 1998
W3C working draft. Available at http://www.w3.org/TR/WD-xml-names.

[PICS]
"PICS 1.1 Label Distribution -- Label Syntax and Communication Protocols", 31 October 1996, T. Krauskopf, J. Miller, P. Resnick, W. Trees
Available at http://www.w3.org/TR/REC-PICS-labels-961031

[RFC1738]
"Uniform Resource Locators", T. Berners-Lee, L. Masinter, and M. McCahill, December 1994.
Available at ftp://ftp.isi.edu/in-notes/rfc1738.txt.

[RFC1766]
"Tags for the Identification of Languages", H. Alvestrand, March 1995.
Available at ftp://ftp.isi.edu/in-notes/rfc1766.txt.

[RFC1808]
"Relative Uniform Resource Locators", R. Fielding, June 1995.
Available at ftp://ftp.isi.edu/in-notes/rfc1808.txt.

[RFC2045]
"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed and N. Borenstein, November 1996.
Available at ftp://ftp.isi.edu/in-notes/rfc2045.txt. Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.

[SMPTE]
"Time and Control Codes for 24, 25 or 30 Frame-Per-Second Motion-Picture Systems - RP 136-1995". Society of Motion Picture & Television Engineers.

[URI]
"Uniform Resource Identifiers (URI): Generic Syntax and Semantics", T. Berners-Lee, R. Fielding, L. Masinter, 4 March 1998.
Available at http://www.ics.uci.edu/pub/ietf/uri/draft-fielding-uri-syntax-02.txt. This is a work in progress that is expected to update [RFC1738] and [RFC1808].

[XML10]
"Extensible Markup Language (XML) 1.0", T. Bray, J. Paoli, C.M. Sperberg-McQueen, editors, 10 February 1998.
Available at http://www.w3.org/TR/REC-xml/

Appendix

Extending SMIL 1.0

(non-normative)

In the future, SMIL 1.0 may be extended by another W3C recommendation, or by private extensions.

For these extensions, it is recommended that the following rules are obeyed:

- All elements introduced in extensions must have a "skip-content" attribute (defined in Section 3.3.1) if it should be possible that their content is processed by SMIL 1.0 players.
- Private extensions must be introduced using the syntax of the XML namespace specification.

It is recommended that SMIL 1.0 players are prepared to handle documents that contain extension that obey these two rules.

Extensions should be handled using an XML namespace mechanism, once such a mechanism becomes a W3C recommendation. In the rest of the section, the syntax and semantics for XML namespaces defined in the W3C note [NAMESPACE] will be used for demonstration purposes only.

The following cases can occur:

1. The document contains a namespace declaration for the SMIL 1.0 specification that defines an empty prefix. In this case, non-SMIL 1.0 elements and attributes are only allowed in a document if they are declared using an XML namespace. The document may not contain a document type declaration for SMIL 1.0. If it does, it is invalid.
   In the following example, the element "new:a" is a legal extension. The elements "mytags:a" and "b" are syntax errors, since they are not declared using an XML namespace.

```
<?xml:namespace ns="http://www.acme.com/new-smil" prefix="new" ?>
<?xml:namespace ns="http://www.w3.org/TR/PR-smil" ?>
<smil>
  <body>
    <par>
      <new:a>
      ...
      </new:a>
      <mytags:a ... />
        ...
      </mytags:a>
      <b>
       ...
      </b>
    </par>
   </body>
 </smil>
```

2. The document contains no document type declaration, it contains a document type declaration for a SMIL version higher than 1.0, or it contains a namespace declaration for a SMIL specification with a version higher than 1.0. For a SMIL 1.0 player to be able to recognize such a namespace declaration, it is recommended that the URI of future SMIL versions starts with http://www.w3.org/TR/REC-smil, and is followed by more characters which may for example be a version number.
   In this case, a SMIL 1.0 player should assume that it is processing a SMIL document with a version number higher than 1.0.
   The following cases can occur:
   Unknown element
   Unknown elements are ignored
   An unknown element may contain content that consists of SMIL 1.0 elements. Whether such content is ignored or processed depends on the value of the "skip-content" attribute. If the attribute is set to "true", or the attribute is absent, the content is not processed. If it is set to "false", the content is processed.
   Content in Element that was declared "Empty"
   A future version of SMIL may allow content in elements that are declared as "empty" in SMIL 1.0.
   Whether this content is ignored or not depends on the value of the "skip-content" attribute of the formerly empty element. If the attribute is set to "true", the content is not processed. If it is set to "false", the content is processed.
   Unknown Attribute

Unknown attributes are ignored.

Unknown Attribute Value

Attributes with unknown attribute values are ignored.

3. The document contains a document type declaration for SMIL 1.0. In this case, it may not contain any non-SMIL 1.0 elements, even if they are declared using XML namespaces. This is because such extensions would render the document invalid.

Using SMIL 1.0 as an Extension

When the XML namespace mechanism is used to include SMIL elements and attributes in other XML-based documents, it is recommended to use the following namespace identifier:

```
http://www.w3.org/TR/REC-smil
```